**#include <bits/stdc++.h>**

**using namespace std;**

**#define M 1000000007**

**bool isFeasible(int i, int j, int cnt, int k) {**

**return (k+1-(j-i) - cnt) > 0;**

**}**

**void solve() {**

**// Iron Magnet Problem**

**int n, k;**

**cin>>n>>k;**

**string str;**

**cin>>str;**

**int prefCondArr[n];**

**memset(prefCondArr, 0, sizeof(prefCondArr));**

**if(str[0] == ':') {**

**prefCondArr[0] = 1;**

**} else {**

**prefCondArr[0] = 0;**

**}**

**for(int i=1;i<n;i++) {**

**prefCondArr[i] += prefCondArr[i-1];**

**if(str[i] == ':') {**

**prefCondArr[i] += 1;**

**}**

**}**

**queue<int>magnets, irons;**

**int ans = 0;**

**for(int i=0;i<n;i++){**

**if(str[i] == 'M') {**

**int flag = 0;**

**while(!irons.empty()) {**

**int idx = irons.front();**

**int cnt = 0;**

**if(idx == 0) {**

**cnt += prefCondArr[i];**

**} else {**

**cnt += prefCondArr[i] - prefCondArr[idx -1];**

**}**

**if(isFeasible(idx, i, cnt, k)) {**

**ans++;**

**flag = 1;**

**irons.pop();**

**break;**

**} else {**

**irons.pop();**

**}**

**}**

**if(flag == 0) {**

**magnets.push(i);**

**}**

**} else if(str[i] == 'X') {**

**while(!irons.empty()) {**

**irons.pop();**

**}**

**while(!magnets.empty()) {**

**magnets.pop();**

**}**

**} else if(str[i] == 'I') {**

**int flag = 0;**

**while(!magnets.empty()) {**

**int idx = magnets.front();**

**int cnt = 0;**

**if(idx == 0) {**

**cnt += prefCondArr[i];**

**} else {**

**cnt += prefCondArr[i] - prefCondArr[idx -1];**

**}**

**if(isFeasible(idx, i, cnt, k)) {**

**ans++;**

**flag = 1;**

**magnets.pop();**

**break;**

**} else {**

**magnets.pop();**

**}**

**}**

**if(flag == 0) {**

**irons.push(i);**

**}**

**}**

**}**

**cout<<ans<<endl;**

**}**

**void solve1(){**

**int n, k;**

**cin>>n>>k;**

**vector<int>arr(n);**

**for(int i=0;i<n;i++){**

**cin>>arr[i];**

**}**

**int res[n];**

**memset(res, 0, sizeof(res));**

**stack<int>stk;**

**for(int i=0;i<n;i++){**

**if(stk.empty() or arr[i]>=arr[stk.top()]) {**

**stk.push(i);**

**} else {**

**while(!stk.empty() and arr[i] < arr[stk.top()]) {**

**res[stk.top()] = (i - stk.top() + 1);**

**stk.pop();**

**}**

**stk.push(i);**

**}**

**}**

**while(!stk.empty()){**

**res[stk.top()] = 1;**

**stk.pop();**

**}**

**int ans = 1;**

**for(int i=0;i<n;i++){**

**ans = (1ll \* ans \* res[i])%M;**

**}**

**cout<<ans<<endl;**

**}**

**int main() {**

**int t;**

**t = 1;**

**while(t--){**

**solve1();**

**}**

**return 0;**

**}**